Information Sciences 325 (2015) 355–374

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Predicting the popularity of micro-reviews: A Foursquare case study

Marisa Vasconcelos*, Jussara M. Almeida, Marcos André Gonçalves

*Department of Computer Science, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, CEP 31270-010 Belo Horizonte MG, Brazil*

ARTICLE INFO

*Article history:*
Received 24 October 2014
Revised 27 March 2015
Accepted 3 July 2015
Available online 8 July 2015

*Keywords:*
Popularity prediction
Content filtering
Location-based social networks
Micro-reviews

ABSTRACT

We tackle the problem of predicting the future popularity level of micro-reviews, focusing on Foursquare tips, whose high degree of informality and briefness offer extra difficulties to the design of effective popularity prediction methods. Such predictions can greatly benefit the future design of content filtering and recommendation methods. Towards our goal, we first propose a rich set of features related to the user who posted the tip, the venue where it was posted, and the tip's content to capture factors that may impact popularity of a tip. We evaluate different regression and classification based models using this rich set of proposed features as predictors in various scenarios. As fas as we know, this is the first work to investigate the predictability of micro-review popularity (or helpfulness) exploiting spatial, temporal, topical and, social aspects that are rarely exploited conjointly in this domain.

## 1. Introduction

Nowadays, the social Web allows people to interact and freely express opinions on products, services or companies in real-time and in large-scale. More and more people base their buying decisions on online reviews written by others [8]. Yet, with the diffusion of smartphones, new services were created targeting mainly social networking users, who spend most of their time accessing information through mobile applications. In this environment, the communication is usually briefer mainly because of the limited amount of information that can be displayed on the mobile screen. This limitation also influenced the creation of new review services (Foursquare, Google+ Local) and the expansion of traditional desktop services to the mobile environment (Yelp, Trip Advisor). In these services, users write *micro-reviews* or *tips*, which are typically much more concise (up to 200 characters), often written while the information is still fresh in the user's mind, and may contain much more subjective and informal content. We here refer to such micro-reviews as simply *tips*.

Accurate tip popularity predictions can drive the design of automatic tip filtering and recommendation schemes, which in turn can help users find tips that are potentially more valuable more easily. Business owners may also benefit from such predictions as they are able to more quickly identify (and fix) aspects of their services or products that may affect revenues most.

However, as in longer review systems, the number of tips on a single product or service may be large and vary greatly in quality [22,26], which makes it hard for users to find helpful reviews. To support that task, many websites allow users to evaluate reviews. Tips, in particular, are often rated by other users clicking on a "like" mark. The number of "likes" received by a tip can then be seen as an estimate of its helpfulness. It can also be used as an estimate of the tip popularity, as it provides a lower bound

* Corresponding author. Tel.: +55 31 3409 5860; fax: +55 31 3409 5858.
*E-mail addresses:* marisav@dcc.ufmg.br, marisavas@gmail.com (M. Vasconcelos), jussara@dcc.ufmg.br (J.M. Almeida), mgoncalv@dcc.ufmg.br (M.A. Gonçalves).

http://dx.doi.org/10.1016/j.ins.2015.07.001
0020-0255/© 2015 Elsevier Inc. All rights reserved.

on the number of people who actually read the tip. Yet, this feedback is usually very sparse. Moreover, ranking tips based only on popularity votes is not useful for promoting recently posted reviews, with a few or no votes, which are doomed to be outranked by older reviews that have already received more votes, and thus lose visibility.

This problem has already inspired a series of studies attempting to automatically predict the quality and helpfulness of online reviews [21,27,41]. However, these efforts focused mostly on textual or content related features (e.g., review length, readability) [21,27,41], which are more suitable for verbose and more formally structured reviews. Moreover, the lack of a "like" does not imply that a tip was not helpful or interesting, as it may not have been seen by any user. This further contributes to make the automatic popularity prediction much harder than in systems that offer a rating scale (e.g., 1–5).

In this article, we study the problem of predicting the popularity *level* a tip will achieve at a future target time, where the popularity level is defined based on the total number of likes the tip receives until that time. More importantly, we focus on predicting the popularity or helpfulness of an individual micro-review exploiting *spatial, temporal, topical and social* aspects that are rarely *conjointly* exploited. To that end, we collected a large and comprehensive dataset of Foursquare tips, covering over 6 million tips and 5 million likes, posted by more than 1.8 million users. Thus, an important contribution of our work is to make this rich dataset available to the research community. Our investigation tackles the following three questions:

*Q1: Which are the most important factors for predicting the popularity of Foursquare tips as soon as it is posted?* We here identify three key entities related to the Foursquare system that may impact a tip's popularity: the user who posted the tip, the venue where it was posted, and its content. We investigate the potential benefits from exploiting these aspects to predict *at posting time* the popularity level of a given tip at a future time. To that end, we first propose a rich set of features related to these three entities, and then evaluate several regression and classification models as well as various feature subsets as predictor variables.

*Q2: To what extent can we improve prediction by monitoring an early period of the tip in the system? How do the prediction models behave as we predict further into the future? Ultimately, how does tip popularity evolve over time?* In Q1, we focused on the hardest prediction scenario, i.e., prediction at posting time, when the only information about the tip consists of its content and historical patterns related to the user and the venue associated with it. In Q2, we assess to what extent prediction accuracy can be improved if, before predicting the popularity of a tip, we monitor the tip for a short period, gathering measures of how its popularity is evolving. Such early popularity measures are then added as predictors to our models. We compare our solutions against three state-of-the-art prediction models that also exploit such early popularity measures [35,38]. We also investigate how far into the future we can predict tip popularity with reasonable accuracy, that is, we analyze how robust our prediction models are when we perform long-term predictions. To motivate these analyses, we first investigate how tip popularity evolves over time.

*Q3: Can we improve prediction accuracy by building specialized models?* We here investigate whether factors related to a specific geographic region (e.g., city) or a type of venue impact a tip's popularity. To that end, we build specialized models using only tips posted in a specific city or in a specific venue category, and compare such models with the single general model.

In sum, the key contributions of this article are: (1) the identification and effectiveness investigation of a rich set of features that influence tip popularity (Q1), (2) a thorough evaluation of the relative performance of state-of-the-art regression and classification techniques in a domain (tip popularity prediction) where they have not been applied before, jointly with the selected features, in various prediction scenarios (Q1 and Q2), and (3) an investigation of the benefits from building specialized models (Q3). This work is a follow up on our prior study of tip popularity prediction [43], which addressed only Q1. We here build on it by introducing Q2 and Q3. In fact, we are unaware of similar work exploiting the temporality issue in the prediction of the popularity level of micro-reviews (research question Q2). Similarly, model specialization (research question Q3) is rarely performed under the spatial, social and topical dimensions we exploit in our investigation.

Next, we first discuss related work in Section 2. We introduce our prediction models and the features used by them in Section 3. A brief analysis of selected features is presented in Section 4.2. We discuss the results of prediction at posting time (Q1) in Section 5. We then investigate tip popularity temporal dynamics, addressing Q2, in Section 6. The impact of model specialization (Q3) is discussed in Section 7. Section 8 offers conclusions and directions for future work.

## 2. Related work

The popularity of a tip, estimated by the number of likes received, can also be seen as an estimate of the tip's helpfulness and quality, as it captures the number of people who found the tip useful. Thus, our work is related to two groups of studies: quality assessment of user generated content, and popularity prediction of online content. We review prior efforts in these two directions next.

### 2.1. Quality assessment of user generated content

Various prior studies focused on analyzing the quality of user generated content, including the quality of Wikipedia articles, video or news comments, and answers on community question answering forums.

For example, Dalip et al. [11] used Support Vector Regression (SVR) to estimate the quality of Wikipedia articles using features related to the text structure, citation network, and article revision history. Siersdorfer et al. [36] proposed a model that uses a term-based representation of YouTube comments (TF-IDF) to automatically classify them as likely to obtain a high overall rating or not. Similarly, Hsu et al. [9] exploited SVR to rank user comments on Digg based on their quality, using various features such as the comment's posting time, number of articles submitted, and comment length. Focusing on user reputation in a comment rating environment (e.g., Yahoo! News), Chen et al. [7] showed that the quality of a comment judged editorially is almost uncorrelated with the ratings that it receives, but it can be predicted using standard text features (e.g., length, spelling, and readability scores). Finally, Momeni et al. [30] developed a classifier for predicting useful comments on YouTube and Flickr exploiting textual features and features that describe the author's posting and social behavior (e.g., number of links posted, and the size of the author's network).

Others have tackled the problem of predicting if a question will have long lasting value. In that context, Anderson et al. [1] found that features of the answer arrival dynamics shortly after the question was posted can help predict the number of page views that the question will receive. Li et al. [24] proposed a mutual reinforcement-based label propagation algorithm to predict the quality of a question using features of the question's text and the asker's profile.

The task of assessing the helpfulness or quality of a review is more closely related to our present effort. Previous approaches to tackle this problem typically employ classification or regression-based methods. For example, Kim et al. [21] used SVR to rank reviews according to their helpfulness, exploiting features related to the textual content of the review and the ratings given by the reviewers. O'Mahony and Smyth [32] proposed a classification-based approach to recommend TripAdvisor reviews, using features related to the user reviewing history and the scores assigned to the hotels. Similarly, Li et al. [28] proposed a nonlinear regression model that uses the reviewer's expertise, the review timeliness, and writing style for predicting the helpfulness of movie reviews. Zhang and Varadarajan [45], in turn, exploited syntactic features (e.g., numbers of nouns, comparatives and modal verbs) to predict the utility of Amazon reviews using SVR and linear regression. Similarly, Ghose and Ipeirotis [14] and Hong et al. [18] approached techniques to predict whether an Amazon review is helpful, while Lee and Choeh [23] used neural networks to assess the helpfulness of Amazon reviews. They found that characteristics of the product such as its list price, its sales rank, and textual characteristics of reviews (e.g., average number of words in a sentence, total number of words, and number of one-letter words) are important for estimating helpfulness.

Those prior studies are based mostly on content features, which are suitable for more verbose and objective reviews, and thus may not be adequate for predicting the popularity of tips, which tend to be more concise and subjective. Moreover, previous studies did not address how helpfulness of reviews as perceived by users (or popularity) *evolve over time*, as we do here. This article extends a recent work of ours [43], which proposed regression and classification algorithms to predict, *at posting time*, the future popularity level of a tip. We here extend this work by: (1) analyzing how the monitoring period and the target time of the prediction affect model accuracy, and (2) assessing the benefits from model specialization.

### 2.2. Popularity prediction of online content

A plethora of prior studies tackled the popularity prediction of online content. We here briefly review some of them. For example, Szabo and Huberman [38] proposed a log-linear model for predicting the long-term popularity of YouTube and Digg content based on early measurements of user accesses. The proposed model uses the popularity acquired by the content during an initial monitoring period to predict its total popularity at a future target time. Later, Pinto et al. [35] modified that model to develop a multivariate regression model by taking the popularity *curve* during the monitoring period into account. One advantage of these two models is that they exploit only information regarding the popularity of the target content during the monitoring period, and thus can be applied to any type of content. They are here used as baselines for comparison against our models for predictions after an initial monitoring period (Section 6).

Aiming at studying the most important factors that influence popularity, Borghol et al. [3] analyzed differences among YouTube videos that have essentially the same content (clones) but different popularity using a multi-linear regression model. Bandari et al. [2] and Hong et al. [17] exploited textual features extracted from messages (e.g., hashtags or URLs), the topic of the message, and user related features (e.g., number of followers) to predict the popularity of news and tweets. Tatar et al. [39] tackled the popularity prediction of news articles based on user comments as a ranking problem. Finally, Castillo et al. [4] proposed a linear regression model to predict the total number of visits of a news article. Unlike news, tweets, and videos, tips are associated with specific venues, and tend to be much less ephemeral as they remain associated with the venue (and thus visible to users) for a longer time. Thus, the popularity of a tip may be affected by features of the target venue as well.

Finally, the only previous investigations of popularity prediction of tips or other micro-reviews we are aware of are our own previous studies [42–44]. As mentioned in the previous section, this work extends [43] by adding various new analyses covering relevant scenarios. In [42], we analyzed the popularity dynamics of groups of micro-reviews, assessing to which extent their popularity ranking changes over time. That work is completely orthogonal to the current article, although both studies have popularity of micro-reviews as main topic. We here investigate the popularity prediction of individual micro-reviews based on their predicted popularity *level*, while in [42] we were interested in *ranking* a group of tips based on their predicted popularity. Moreover, the present effort also differs from [44], where we uncovered different profiles related to how users exploit these features. In comparison with that work, we here focus on a different (though related) problem – the prediction of tip popularity.

## 3. Tip popularity prediction models

In this section, we first define our target problem (Section 3.1), and then introduce the features used to capture key factors that impact tip popularity (Section 3.2) as well as the prediction methods that exploit these features as predictor variables (Section 3.3).

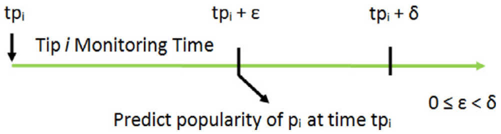


**Fig. 1.** Prediction scenarios.

### 3.1. Problem definition

Our goal is to develop models to predict the *popularity level* of a tip $p_i$, posted at time $tp_i$, at a given future target time $tp_i + \delta$, where the popularity of the tip is estimated by the total number of likes received until $tp_i + \delta$. Thus, $\delta$ defines the prediction window. Moreover, we consider various scenarios where the predictions are performed at time $tp_i + \varepsilon$, where $\varepsilon < \delta$ defines a period after posting time, during which the tip is monitored. Note that $\varepsilon$ may be zero, corresponding to predictions at posting time. Fig. 1 illustrates the prediction scenarios considered in our study.

Towards our goal, we categorize tips into various *popularity levels*, defined based on the number of likes received (see below). As in [2,17,28], we predict the popularity *level*, instead of the exact number of likes, because the latter is harder, particularly given the skewed distribution of tip popularity [44], and the former should be good enough for most purposes.

As in other prediction tasks, the prediction model is learned using a training set, which consists of a subset of tips along with associated information about the users who posted them, the venues where they were posted, and textual characteristics extracted from their content. The learned prediction model is then evaluated using a different set of tips (test set). Both training and test sets are built considering three different types of entities: a set $P = \{p_1, \ldots, p_K\}$ of $K$ tips, a set $U = \{u_1, \ldots, u_N\}$ of $N$ users (tip authors), and a set $V = \{v_1, \ldots, v_O\}$ of $O$ venues. Each tip is represented by a tuple $(p, u, v)$, and each entity $(p, u$, and $v)$ has a set of attributes (or features) associated with it. These features represent the inputs (predictors) associated with a given instance. There are also relationships between these sets of entities: a function $L$: $P \rightarrow V$ maps each tip $p_i$ to a unique venue $v_i$, and an authorship function $A$: $P \rightarrow U$ maps each tip $p_i$ to a unique user $u_i$. For evaluation purposes, each tip $p_i \in P$ is labeled with a numeric value that represents the popularity level of $p_i$ at time $tp_i + \delta$. The values of the features associated with each entity are computed using information available up to the moment when the prediction is done, i.e., up to $tp_i + \varepsilon$.

Thus, given the input data $< p, u, v >$, we want to learn a prediction model $M$ that, for each tip $p_i$ posted at time $tp_i$ and monitored until $tp_i + \varepsilon$, predicts its popularity level at time $tp_i + \delta$. A tip is represented by a $f$-dimensional real vector $\mathbf{p}$ over feature space built from the information in $P$, $U$ and $V$. The model is thus a function $M : \mathbb{R}^f \rightarrow \mathbb{R}$ that maps a tip's feature vector into a numerical popularity level. We note that, in essence, this is a classification problem where each popularity level defines a popularity *class*. Thus we use both terms, namely popularity level and class, interchangeably throughout this paper.

In our experiments, we consider two popularity levels: low and high. Tips that receive at most 4 likes during the period $tp_i + \delta$ have *low popularity*, whereas tips that receive at least 5 likes in the same period have *high popularity*. We note that the availability of enough examples from each class for learning the model parameters impacts the definition of the popularity levels. We also experimented with other numbers of levels (e.g., three) and range definitions, finding, in general, similar results. Alternatively, we tried to group tips using various features and clustering techniques (e.g., k-means, x-means, spectral and density-based clustering). However, as previously observed for quality prediction in question answering services [24], the resulting clusters were not stable (i.e., results varied with the seeds used). This may be due to the lack of discriminative features *specifically for the clustering process*, which does not imply that the features cannot be useful for the prediction task, our main goal here.

Our proposed solution to the aforementioned problem consists in: (1) defining the set of features used to represent the tips, and (2) applying a learning algorithm to predict the popularity level of a given tip instance $p_i$, given $\varepsilon$ and $\delta$. To our knowledge, we are the first to tackle this problem. Thus, identifying which factors related to each key entity of the problem should be considered as input features to our prediction models is a key contribution of this work. As we further discuss in the next section, some of these features are adapted from previous approaches in related domains to the particular context of tip popularity, while others are new contributions that capture particular aspects of the types of content and application under study. Also, as we discuss in Section 3.3, we explore different learning algorithms, notably classification and regression models, to predict tip popularity. The selected techniques have been applied before in other prediction tasks. Our contribution lies mostly in assessing their relative performance in the context of tip popularity prediction in different predictions tasks and scenarios.

### 3.2. Tip features

One key contribution of this work is to identify different factors that may impact tip popularity and thus should be considered in the design of popularity prediction models. Specifically, these factors are captured by means of features used as predictors in our prediction models. We here identify $k = 125$ features related to the three key entities related to our target problem, namely, the user $u_i$ who posted $p_i$, the venue $v_i$ where $p_i$ was posted, and the content of $p_i$. We summarize the selected features as follows:

*User features*: This set of features captures the activities performed by the tip author (e.g., number of tips, number of likes received/given) and her social network (e.g., number of friends/followers, percentage of likes coming from her social network, numbers of tips posted and likes given by her social network). We consider two scenarios when computing user features: (1) all tips posted by the user, and (2) only tips posted by the user at venues of the same category of the venue where $p_i$ was posted. We refer to the latter as *user/cat features*.

*Venue features*: This set of features captures the activities at the venue where $p_i$ was posted and its visibility. Examples of venue features are the number of tips posted at the venue, the number of likes received by those tips, the numbers of check ins and unique visitors, and the venue category.

*Tip's content features*: This set contains features related to the structure, syntactics, readability and sentiment of the tip's content [7,12,30,33]. It includes the numbers of characters, words, and URLs or e-mail addresses in the tip, readability metrics, features based on the Part-Of-Speech tags (e.g., percentages of nouns and adjectives), and features based on psychological dimensions defined by the LIWC dictionary [40]. It also includes three scores – positive, negative and neutral – that capture the tip's sentiment. These scores are computed using SentiWordNet [13], an established sentiment lexical for supporting opinion mining in English texts. In SentiWordNet, each term is associated with a numerical score in the range [0, 1] for positive, negative and objectivity (neutral) sentiment information. We compute the scores of a tip by taking the averages of the corresponding scores over all words in the tip that appear in SentiWordNet. To handle negation, we adapt a technique proposed in [34] that reverses the polarity of words between a negation word ("no", "didn't", etc.) and the next punctuation mark.

The complete list of all 125 tips considered in this work is presented in Table 6 (Appendix). We refer to these features throughout the paper by their feature ID, also shown in the Table. We note that two features, namely number of likes received by tip $p_i$ until time $tp_i + \varepsilon$ and the position of the tip in the ranking of the venue sorted by date in descending order, are used only in prediction scenarios where the monitoring time $\epsilon$ is non-zero (Section 6).

We also note that some of the identified features, such as average number of likes received by all previous tips of user $u_i$ as well as the features related to the content of tip $p_i$, have also been previously explored to analyze the helpfulness of online reviews [21,45] and predict the ratings of (long) reviews [29,36]. Others have been adapted from previous prediction approaches in other domains (e.g., question answering sites, Wikipedia, and YouTube [1,11,12,30,36]. However, many features related to the user who posted the tip, such as the number of venues where the user posted tips previously (feature 2), the total number of likes given by the user (feature 7), as well as features related to the user's social network (features 9–14) are new proposals of this work. Similarly, some features related to the popularity of the venue where the tip was posted (features 30–31) are also new. As will be discussed in Section 5, several of these new features are very important to prediction accuracy.

### 3.3. Prediction methods

As in [21,28,32], we exploit regression and classification techniques in the design of our tip popularity prediction models.

We experiment with two types of regression algorithms. Both produce as output a real value, which is rounded to the nearest integer that represents a popularity level, as done for predicting the helpfulness of movie reviews in [28]. Our first approach is an ordinary least squares (OLS) multivariate linear regression model. It estimates the popularity level of a tip $p_i$ at a given point in time $t$, $\mathcal{R}(p_i, t)$, as a linear function of $k$ predictor variables $x_1, x_2, \ldots, x_k$, i.e., $\mathcal{R}(p_i, t) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_k x_k$. Model parameters $\beta_0, \beta_1, \ldots, \beta_k$ are determined by the minimization of the least squared errors [15] in the training data (see Section 5.1). Given a tip in the test set, popularity prediction is done by using the values of the selected features as predictor variables $x_1, x_2, \ldots, x_k$.

We also consider the more sophisticated Support Vector Regression (SVR) algorithm [15], a state-of-the-art method for regression learning. Unlike the OLS model, SVR does not consider errors that are within a certain distance of the true value (within the *margin*). It also allows the use of different kernel functions, which help solving a larger set of problems, compared to linear regression. We use both linear and radial basis function (RBF) kernels, available in the LIBSVM package [6], as the latter handles non-linear relationships.

Finally, we experiment with the Support Vector Machine (SVM) algorithm [15] to *classify* tips into popularity levels. The goal of SVM is to find an optimal separating hyperplane in the feature space that gives the largest minimum distance to the training examples. Again, we experiment with both linear and RBF kernels.

We note that the three aforementioned techniques have been applied before to prediction tasks in other domains. Our goal here was not to propose a new learning technique but rather investigate how state-of-the-art techniques perform in our target problem, in different scenarios. In particular, we are interested in assessing the cost-benefit ratio of the simpler linear regression method (OLS) compared to the more sophisticated (and also more costly) SVR and SVM techniques.

## 4. Feature analysis

Before evaluating our prediction models, we first describe the dataset used in our experiments (Section 4.1), and briefly analyze some selected features (Section 4.2).

### 4.1. Dataset

We crawled Foursquare using the system API from August to October 2011, collecting data associated with more than 13 million users. We believe that this represents a large fraction of the total user population at the time, which, reportedly, varied from 10 to 15 million between June and December 2011.[1] We made our dataset publicly available for research purposes.[2]

[1] https://foursquare.com/infographics/10million and http://www.socialmedianews.com.au/foursquare-reaches-15-million-users/.

[2] http://vod.dcc.ufmg.br/traces/foursq/data.html.

**Table 1**
Overview of selected features.

| Type | Feature | Min | Mean | Max | CV |
|---|---|---|---|---|---|
| **User** | Number of tips posted | 1 | 3.72 | 5791 | 3.25 |
| | Number of likes received | 0 | 3.13 | 208,619 | 63.40 |
| | Median number of likes received | 0 | 0.48 | 657.0 | 2.77 |
| | Mean number of likes received | 0 | 0.58 | 858.10 | 2.70 |
| | Social network (SN) size | 0 | 44.79 | 318,890 | 15.95 |
| | Percentage of likes from SN | 0 | 0.71 | 1 | 0.43 |
| **Venue** | Number of tips | 1 | 2.13 | 2419 | 2.23 |
| | Number of likes | 0 | 1.80 | 7103 | 11.60 |
| | Median number of likes | 0 | 0.45 | 390 | 2.81 |
| | Number of check ins | 0 | 217.33 | 484,683 | 6.18 |
| | Number of visitors | 0 | 87.35 | 167,125 | 5.87 |
| **Content** | Number of words | 1 | 10.25 | 66 | 0.78 |
| | Number of characters | 1 | 59.78 | 200 | 0.75 |
| | Number of URLs | 0 | 0.02 | 9 | 8.27 |

Our complete dataset contains almost 16 million venues and over 10 million tips. However, to avoid introducing biases towards very old or very recent tips, we restricted our analyses to tips and likes created between January 1st 2010 and May 31st 2011. This filter left us with over 6 million tips and almost 5.8 million likes, posted at slightly more than 3 million venues by more than 1.8 million users. We note that 34% of the tips in our analyzed dataset received, during the considered period, at least one like.

We should stress that the research problem we address in this article is a brand new one, which makes very hard to find other datasets with similar characteristics to our Foursquare collection. Foursquare is the currently most popular and largest LBSN micro-review system, having been established since 2009. Only recently other systems such as Yelp (in 2013) and Facebook Places tips (in 2015) have been established in this area. Because of this, they have not achieved enough critical mass in terms of data size and representativeness to allow the kind of experimentation we have performed here. Moreover, our Foursquare dataset is very diverse and rich, having characteristics that are currently very hard (if not impossible) to obtain in other similar systems, such as the timestamp of likes received by each tip, which is not available any more in Foursquare. This information was essential to perform our temporal analysis and to make our experimental methodology very comprehensive.

### 4.2. *Feature characterization*

Table 1 presents statistics of selected features for users and venues with at least one tip in our dataset. The table shows, for each feature, minimum, mean, maximum values and coefficient of variation (CV), which is the ratio of the standard deviation to the mean. We note that most features, but particularly the number of likes received by tips previously posted by the user and at the venue, have very large CV's, reflecting their heavy tailed distributions, as observed in [44].

Indeed, most users posted a very few tips and/or received a few likes while most tips and likes were posted by a few users. For example, 46% and 48% of the users posted only one tip and received only one like, respectively. In contrast, only 1499 and 2318 users posted more than 100 tips and received more than 100 likes, respectively. These heavy tailed distributions suggest that tips may experience the rich-get-richer effect. Moreover, the median number of likes per user is, on average, only 0.48. Thus, many users have this feature equal to 0. This will impact our prediction results, as discussed in Section 5.

The Spearman's correlation coefficient ($\rho$), which is a non-parametric measure of statistical dependence between two variables [46], computed over the numbers of tips and likes received by each user is moderate ($\rho = 0.54$), and between the numbers of tips and likes given by each user is even lower ($\rho = 0.37$). Thus, in general, users who tip more do not necessarily receive or give more feedback.

We also find that the social network is quite sparse among users who post tips: a user has only 44 friends/followers, on average, while 37% of the users have at most 10 friends (followers), although the maximum reaches 318,890. Moreover, the fraction of likes coming from the user's social network tends to be reasonably large (70% on average). Thus, the user's social network does influence the popularity of tips.

There is also a heavy concentration of activities (check ins, visitors, tips, and likes) on a few venues. The maximum number of likes per venue exceeds 7000, but the average is only 1.8. The correlation between the number of check ins (or the number unique visitors) and the number of tips of the venue is moderate (around 0.52). Thus, to some extent, popular venues tend to attract more tips, although this is not a strong trend. The correlation between the number of tips and the total number of likes per venue is also moderate ($\rho = 0.5$). Thus, in general, tipping tends to be *somewhat* effective in attracting visibility (likes) to a venue. The same correlation exists between the total number of likes and the number of check ins ($\rho = 0.5$), but we cannot infer any causality relationship. However, the correlation between the median number of likes and the number of check ins is weaker ($\rho = 0.3$), implying that not all tips posted at the same venue receive comparable number of likes. This is probably due to various levels of interestingness of those tips, but might also reflect the rich-get-richer effect.

Finally, regarding content features, we find that most tips are very short, with, on average, around 60 characters and 10 words, and at most 200 characters (limit imposed by the application) and 66 words. Moreover, the vast majority (98%) of the tips carry no URL or e-mail address, and we find no correlation between the size of the tip and the number of likes received by it ($\rho < 0.07$), as one might expect.

## 5. Tip popularity prediction at posting time (Q1)

We start our evaluation by considering predictions done at the time the tip is posted. That is, we fix $\varepsilon = 0$ and evaluate our proposed models to predict at time $tp_i$ the popularity level that tip $p_i$ will achieve at time $tp_i + \delta$. We also set $\delta$ equal to 1 month, and leave the evaluation for other values of $\delta$ and $\varepsilon$ to Section 6). Next, we present our experimental setup (Section 5.1), and then discuss our main results (Section 5.2).

### 5.1. Experimental setup

Our methodology consists on dividing the available data into training and test sets, learning model parameters using the training set, and evaluating the learned model in the test set. We split the tips chronologically into training and test sets, rather than randomly, to avoid including in the training set tips that were posted after tips for which predictions will be performed. We build 5 runs, thus producing 5 results, by sliding the window containing training and test sets.

To learn the models, we consider the most recent tip posted by each user in the training set as *candidate* for popularity prediction, and use information related to the other tips to compute the features used as predictor variables. All tips in the test set are taken as candidates for prediction, and their feature values are computed using all information (e.g., tips, likes) available until the tip's posting time (i.e., when the prediction is performed). For each candidate in both training and test sets, we compute the feature values by first applying a logarithm transformation on the raw numbers to reduce their large variability, normalizing the results between 0 and 1. Moreover, in order to have enough historical data about users who posted tips, we consider only users who posted at least 5 tips. To control for the age of the tip, we only consider tips that were posted at least 1 month before the end of training/test sets. We also focus on tips written in English, since the tools used to compute some textual features are available only for that language. After applying these filters, we ended up with 707,254 tips that are *candidates* for prediction.

The distribution of candidate tips into the popularity levels is very skewed, with 99.5% of the tips with low popularity. Such imbalance, particularly in the training set, poses great challenges to regression/classification accuracy. Indeed, initial results produced using all available data were very poor. Instead, a widely used strategy to cope with the effects of class imbalance in the training data is under-sampling [16]. Suppose there are $n$ tips in the smallest category in the training set. We produce a balanced training set by randomly selecting equal sized samples from each category, each with $n$ tips. Note that under-sampling is performed only in the training set. The test set remains unchanged. Because of the random selection of tips for under-sampling, we perform this operation 5 times for each sliding window, thus producing 25 different results, in total. The results reported in the next sections are averages of these 25 results, along with corresponding 95% confidence intervals.

OLS model parameters are defined by minimizing the least squared errors of predictions for the candidate tips in the training set. This can be done as their popularity levels at $tp_i + \delta$ are known. Best values for SVM and SVR parameters are defined in a similar manner, using cross-validation within the training set.

We compare the accuracy of the OLS, SVR and SVM models against that of a simple (baseline) strategy that uses the median number of likes of tips previously posted by user $u$ as an estimate of the number of likes that a new tip posted by $u$ will receive (and thus its popularity level). Note that state-of-the art prediction methods, such as those proposed in [35,38] cannot be applied in the scenario considered here, as they exploit early popularity measures as predictors, and thus require $\varepsilon > 0$ . We will consider them as baselines in the next section.

We evaluate the prediction models using precision and recall of each class (i.e., popularity level), as well as macro-averaged precision and macro-averaged recall. The *precision* of class $i$ is the ratio of the number of tips correctly assigned to $i$ to the total number of tips *predicted* as $i$, while the *recall* is the ratio of the number of correctly classified tips to the *actual* number of tips in class $i$. The macro-averaged precision (recall) is the average precision (recall) computed across both classes.

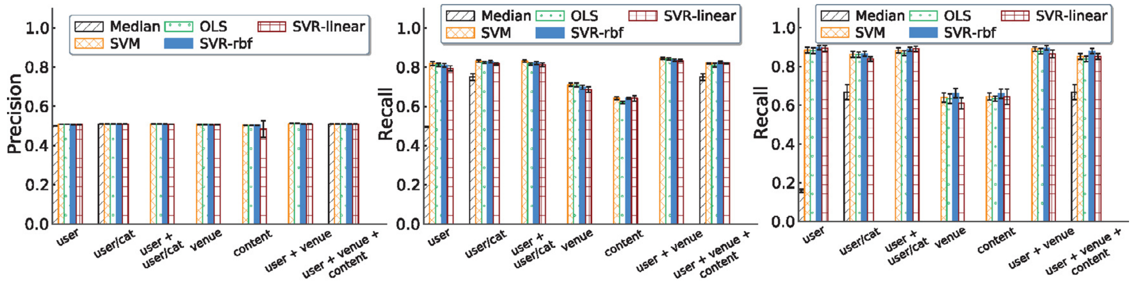
### 5.2. Prediction results

We now discuss representative results for predictions performed at posting time. We first investigate how the sets of features related to the three central entities – user, venue and tip – affect the prediction of tip popularity (Section 5.2.1). We then analyze the importance of each feature individually (Section 5.2.2), and assess the presence of multicollinearity in our prediction models (Section 5.2.3).

#### 5.2.1. Analysis of the groups of features

Fig. 2 (a) and (b) shows macro-averaged precision and recall, along with 95% confidence intervals, for 30 prediction strategies that emerge from the combination of five prediction algorithms with alternative sets of predictor variables. In particular, for OLS, SVR (linear and RBF kernels) and SVM[3] algorithms, we consider the following sets of predictors: only user features, only venue

[3] We show only results for the linear kernel as results produced with the RBF kernel are similar.



(a) Macro-Averaged Precision (b) Macro-Averaged Recall (c) Recall for High Popularity

**Fig. 2.** Effectiveness of alternative prediction models and sets of predictor variables.

features, only content features, all venue and user features, all user, venue and content features. We also consider only user features restricted to the category of the venue where the tip was posted (user/cat features). For predictions using the median number of likes of the user, here referred to as median strategy, we compute this number over all tips of the user and only over the tips posted at venues of the same (target) category.[4] The significance of the results was assessed using both one-way ANOVA and Kruskal–Wallis tests [46]) with 95% confidence.

Fig. 2(a) shows that there is little difference in macro-averaged precision across the prediction algorithms, except for SVR with linear kernel using only content features, which performs worse. The same degradation of SVR is observed in terms of macro-averaged recall (Fig. 2(b)). For that metric, the superiority of SVM, SVR and OLS over the simpler median strategy is clear. For any of those strategies, the best macro-averaged recall was obtained using user and venue features, with gains from 1% (SVR-RBF) up to 36% (OLS) over the other sets of predictors. Comparing the different techniques using user and venue features, we see small but statistically significant gains in macro-averaged recall for SVM and OLS over SVR (up to 1.15% and 0.84%, respectively) while OLS and SVM produce statistically tied results. However, OLS is much simpler than both SVM and SVR, as will be discussed below.

We also note that recall results are higher than precision results in Fig. 2(a) and (b). This is because the precision for the smaller class (high popularity) is very low even with a high recognition rate (large number of true positives). Recall that under-sampling is applied only to the training set, and thus class distribution in the test set is still very unbalanced and dominated by the larger class (low popularity). Thus, even small false negative rates for the larger class results in very low precision for the other (smaller) class. For that reason, towards a deeper analysis of the prediction strategies, we focus primarily on the recall metric and discuss separate results for each class, with special interest in the high popularity class. The focus on recall is particularly interesting for tip filtering and recommendation tools where users (regular users or venue/business owners) may want to retrieve most of the potentially popular tips at the possible expense of some noise (tips in the low popularity class).

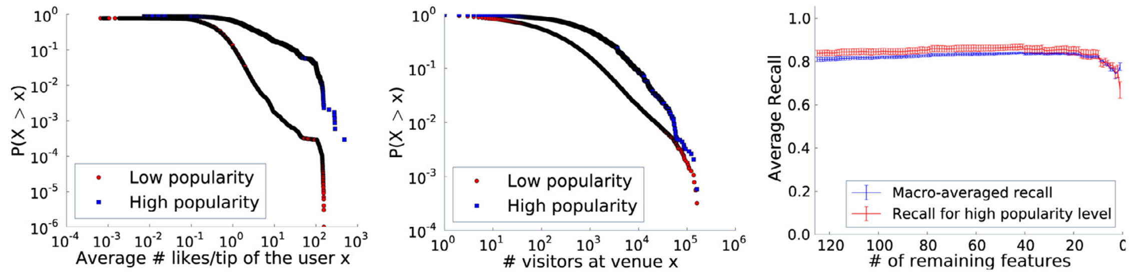
Fig. 2(c) shows average recall of tips for the high popularity level. Note that, for any prediction model, there are no significant differences in the *recall* of high popularity tips across the various sets of predictors, provided that user features are included. Moreover, the use of venue features jointly with user features improves the *precision* of the high popularity class (figure omitted) in up to 46% (35% for the OLS algorithm). That is, adding venue features reduces the amount of noise when trying to retrieve potentially popular tips, with no significant impact on the recall of that class, and is thus preferable over exploring only user features. Note also that including content features as predictors lead to no further (statistically significant) gains.

Comparing OLS, SVR and SVM with user and venue features as predictors, we find only limited gains (if any) of the more sophisticated SVR and SVM algorithms over the simpler OLS strategy. In particular, SVR (with RBF kernel) leads to a small improvement (2% on average) in the recall of the high popularity class, being statistically tied with SVM. Yet, in terms of precision of that class, OLS outperforms SVR by 13.5% on average, being tied with SVM. We note that SVR tends to overestimate the popularity of a tip, thus slightly improving the true positives of the high popularity class (and thus its recall) but also increasing the false negatives of the low popularity class, which ultimately introduces a lot of noise into the predictions for the high popularity class (hurting its precision).

We note that the limited gains in recall of SVR over OLS come at the expense of a much longer model learning process, for a fixed training set. Indeed, we measured the time required to train each model on a 2.40 GHz Intel Xeon machine running Ubuntu Linux 12.04. The average time per tip in the training set to learn the OLS model (i.e., normalized training time) is 0.55 ± 0.01 ms. In contrast, the average normalized training times for SVM and SVR (with RBF kernel) are 6.26 ± 1.03 and 52.60 ± 2.49 ms, respectively. That is, on average SVM takes 11 times longer than OLS in the model training phase. SVR (with RBF kernel) takes even longer (96 times longer, on average). SVR (with linear kernel), in turn, has a normalized training time that is orders of magnitude longer than that of the three other methods, reaching 29.84 ± 0.29 s, on average.

Thus, even though the time to perform predictions is basically the same for all methods (as it consists in applying a learned function over the feature values), they differ greatly in terms of the training time. Short training times are desirable to enable more frequent updates of the model to reflect changes in the features. Thus, if we consider not only prediction accuracy but also

[4] The figures show results of the median prediction model only for the user and user/cat sets of predictor variables since they are the same for the other sets.



(a) Best user feature

(b) Best venue feature

(c) Recall when removing one feature at a time

**Fig. 3.** Importance of individual features for tip popularity.

model training time, we find that the simpler OLS model produces results that, from a practical perspective, are very competitive to (if not better than) those obtained with SVM and SVR.

### 5.2.2. *Effects of individual features*

We now analyze the relative importance of each feature for prediction accuracy by using the Information Gain (IG) feature selection technique [12] to rank features according to their discriminative capacity for the prediction task. We here use the OLS method with the complete set of features.

The three most important features according to IG are related to the tip's author. They are the average, total and standard deviation of the number of likes received by tips posted by the user. Thus, the feedback received on previous tips of the user is the most important factor for predicting the popularity level of her future tips. Fig. 3(a), which shows the complementary cumulative distributions of the best of these features (average number of likes) for tips in each class, clearly indicates that it is very discriminative of tips with different (future) popularity levels. Features related to the social network of the tip's author are also important. The number of friends/followers of the author and the total number of likes given by them occupy the 4th and 9th positions of the ranking, respectively. Moreover, we find that authors of tips that achieve high popularity tend to have more friends/followers.

The best venue feature, which occupies the 6th position of the ranking, is the number of unique visitors (Fig. 3(b)). Moreover, total number of check ins (7th position), venue category (8th position) and total number of likes received by tips posted at the venue (11th position) are also very discriminative, appearing above other user features, such as number of tips (19th position), in the ranking.

Finally, we extend our evaluation of the importance of different sets of features by assessing the accuracy of the OLS strategy as we remove one feature at a time, in increasing order of importance given by the Information Gain. Fig. 3(c) shows the impact on the macro-averaged recall and on the average recall of the high popularity class as each feature is removed, starting with the complete set of user, venue and content features. For example, the second point in each curve shows results after removing the least discriminative feature (number of common misspellings per tip). Note that the removal of many of the least discriminative features has no significant impact on recall, indicating that these features are redundant. Accuracy loss is observed only after we start removing features in the top-10 positions. Among those, the largest losses are observed when the number of check ins at the venue and the size of the user's social network are removed, which reinforces the importance of venue and social network features to the prediction task. In sum, using the top 10 most important features produces predictions that are as accurate as those of using the complete set of features is used. Next, we test whether multicollinearity exists among different predictors, which could affect the accuracy of the OLS model.

### 5.2.3. *Multicollinearity analysis*

Multicollinearity occurs in linear regression models when two or more predictor variables are linearly correlated. Although the SVM and SVR methods are capable of handling a high degree of collinearity [19], the OLS model may be severely impacted [20], as multicollinearity can increase the variance of the OLS coefficient estimates, degrading model predictability.

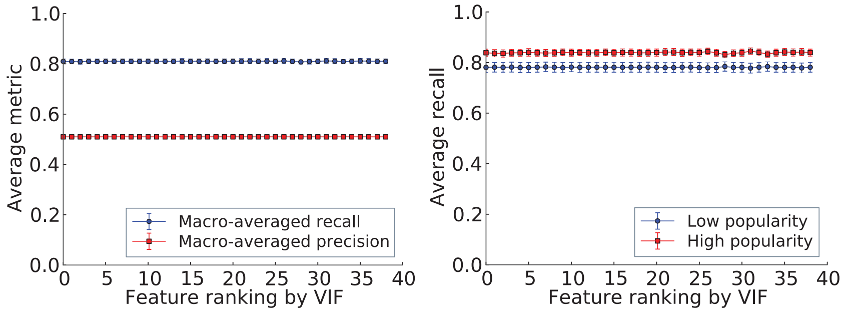
There are several methods to test multicollinearity. We here test whether our OLS prediction model is impacted by multicollinearity using two methods: variance inflation factors (VIF) [37] and tolerance [31]. The VIF for a predictor $k$, $VIF_k$, indicates whether there is a strong linear association between $k$ and all the remaining predictors. If multicollinearity exists, the variance of an estimated regression coefficient $b_i$ is inflated by $VIF_k$ because of the correlation among the predictor variables in the model. The variance inflation factor for the $i$th predictor is computed as $VIF_i = \frac{1}{1-R_i^2}$, where $R_i^2$ is the coefficient of determination obtained by a regression model built using the $i$th predictor as response variable and the remaining predictors as inputs. Stevens [37] recommends a heuristic VIF greater than 10 as an indication of multicollinearity requiring correction. The tolerance is measured as $1 - R_i^2$. A tolerance of less than 0.10 indicates a multicollinearity problem [31].

We compute the VIFs and tolerances for all features of our complete OLS model. Table 2 lists the features with VIF values above 10, with corresponding average VIF and tolerance values (and 95% confidence intervals). Several features are collinear

**Table 2**
Features with high collinearity with at least one other feature.

| ID[a] | VIF | Tolerance | ID[a] | VIF | Tolerance | ID[a] | VIF | Tolerance |
|---|---|---|---|---|---|---|---|---|
| 36 | 507.83 ± 44.80 | 0.004 ± 0.00 | 42 | 34.33 ± 1.74 | 0.06 ± 0.00 | 1 | 16.52 ± 2.07 | 0.13 ± 0.01 |
| 25 | 215.71 ± 15.94 | 0.01 ± 0.00 | 64 | 32.69 ± 2.29 | 0.06 ± 0.01 | 16 | 16.52 ± 2.10 | 0.13 ± 0.02 |
| 26 | 101.03 ± 4.78 | 0.02 ± 0.00 | 10 | 27.19 ± 3.47 | 0.08 ± 0.01 | 2 | 15.33 ± 1.67 | 0.13 ± 0.01 |
| 40 | 96.01 ± 9.53 | 0.02 ± 0.00 | 46 | 21.92 ± 0.99 | 0.09 ± 0.00 | 65 | 15.19 ± 0.38 | 0.13 ± 0.00 |
| 39 | 80.60 ± 7.81 | 0.03 ± 0.00 | 76 | 21.36 ± 1.19 | 0.09 ± 0.01 | 8 | 13.95 ± 1.67 | 0.15 ± 0.02 |
| 61 | 59.83 ± 4.17 | 0.03 ± 0.00 | 78 | 21.02 ± 1.24 | 0.09 ± 0.01 | 51 | 12.85 ± 0.60 | 0.15 ± 0.01 |
| 35 | 49.46 ± 3.08 | 0.04 ± 0.00 | 68 | 19.85 ± 1.47 | 0.10 ± 0.01 | 17 | 12.64 ± 0.91 | 0.16 ± 0.01 |
| 53 | 45.92 ± 2.66 | 0.04 ± 0.00 | 82 | 19.10 ± 1.17 | 0.10 ± 0.01 | 54 | 12.56 ± 0.80 | 0.16 ± 0.01 |
| 21 | 44.03 ± 1.25 | 0.05 ± 0.00 | 83 | 18.60 ± 1.21 | 0.11 ± 0.01 | 90 | 12.52 ± 0.58 | 0.15 ± 0.01 |
| 34 | 43.59 ± 2.76 | 0.05 ± 0.00 | 28 | 18.57 ± 1.11 | 0.11 ± 0.01 | 44 | 11.86 ± 0.54 | 0.16 ± 0.01 |
| 20 | 40.11 ± 1.51 | 0.05 ± 0.00 | 43 | 18.53 ± 0.72 | 0.11 ± 0.00 | 31 | 11.84 ± 0.53 | 0.16 ± 0.01 |
| 14 | 37.20 ± 6.03 | 0.06 ± 0.01 | 18 | 18.50 ± 0.57 | 0.11 ± 0.00 | 91 | 10.23 ± 0.43 | 0.19 ± 0.01 |
| 52 | 34.99 ± 2.15 | 0.06 ± 0.00 | 5 | 18.49 ± 0.71 | 0.11 ± 0.00 | | | |

[a] The feature IDs are defined in Table 6.



(a) Macro-Averaged Results

(b) Averaged Recall

**Fig. 4.** Macro-averaged results for OLS after removing each collinear feature.

with at least one other feature in the model. This is not totally unexpected as most collinear features are derived from other features (e.g., total and average number of likes of the user). Note also that some of these features have high information gain.

Next, we perform an experiment with the OLS method, eliminating one collinear feature at a time. We also compare the OLS model without the collinear feature and the original complete model. Fig. 4(a) shows the macro-averaged results, while Fig. 4(b) shows the average recall values for each popularity category. Each point in the $x$-axis represents a scenario when one collinear feature is eliminated: the eliminated feature is identified by its position in the ranking of VIF values (see Table 2), starting with the feature with largest VIF (i.e., feature 36). The first point ($x = 0$) refers to the original method with all features. The figures show that the multicollinearity does not impact any of our metrics, i.e., the gains after eliminating each multicolinear feature are not statistically significant neither for macro-averaged precision and recall nor for average per-class recall.

## 6. Temporal dynamics of tip popularity prediction (Q2)

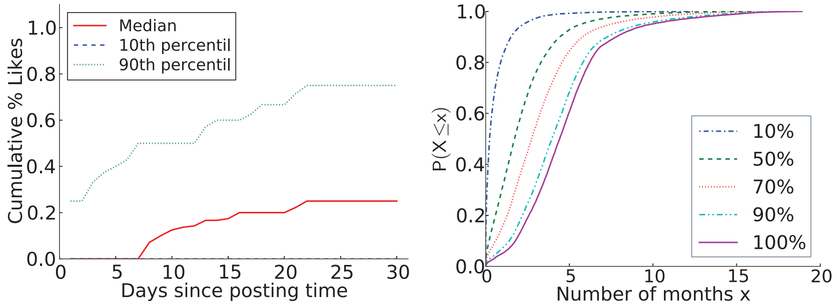
Ideally, one would like to predict the future popularity of a tip immediately after it is posted. This was the scenario considered in the previous section. However, tips may exhibit different popularity evolution patterns, as previously observed for other types of content (e.g., YouTube videos [10,35], Digg news [38], and tweets [17]). By monitoring the tips for a certain (short) time interval ($\varepsilon$ units of time), we may be able to gather useful information about how its popularity is evolving, which may improve prediction accuracy. Similarly, so far we have considered only predictions targeting one month ahead. However, depending on how the popularity of a tip evolves over time, we may be able to make accurate predictions further in the future (i.e., larger values of $\delta$). One concern in this case is that, as we predict further into the future, the information used as predictors may get outdated, hurting prediction accuracy.

In order to address these questions, we first briefly analyze tip popularity evolution in Section 6.1. Next, we assess the impact of varying the monitoring time $\varepsilon$ and the target prediction window $\delta$ on prediction accuracy in Sections 6.2 and 6.3, respectively.

### 6.1. Tip popularity evolution

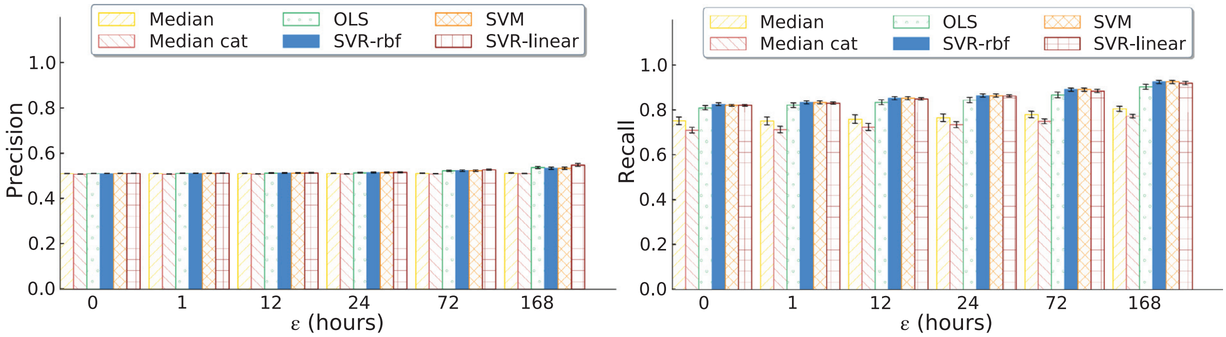
We briefly discuss tip popularity evolution by analyzing how the number of likes received by the most popular tips evolves over time. This number is analyzed as a percentage of the total number of likes of the tip at the time our dataset was collected.

Fig. 5(a) shows how the popularity of the top-10% most popular tips in our dataset evolves over time, during the first month after posting time. The figure shows the curves of 10th and 90th percentiles as well as median percentage of likes. We note that



(a) % likes received in the $1^{st}$ month (b) Time until x% of total likes are received

**Fig. 5.** Cumulative distribution of popularity for the top-10% most popular tips.



(a) Macro-Averaged Precision (b) Macro-Averaged Recall

**Fig. 6.** Macro-averaged results for various monitoring times $\varepsilon$ ($\delta = 1$ month).

10% of the analyzed tips did not receive any like during the first month, and thus the 10th percentile curve coincides with the *x*-axis. In contrast, around half of the most popular tips started receiving likes after 7 days since posting time. Moreover, those same tips received only up to 20% of their total likes in the first month since posting. Indeed, the 90th percentile curve indicates that the vast majority of the tips received at most 75% of their total likes in the first month in the system. These results show that: (1) there is great variability in the popularity evolution across tips, even considering only the top 10% most popular tips, and (2) half of those tips receive a small fraction of their total likes within one month in the system.

We also analyze the time it takes for a tip to receive *X*% of their likes. These results are shown in Fig. 5(b) for various values of *X*. We note that 76% of the tips take at least 3 months to reach 50% of its total observed popularity, which indicates that tips experience a somewhat slow popularity evolution, particularly if compared to other types of content (e.g., news and photos [5,38]).

The great variability in popularity evolution across tips and the somewhat slow popularity dynamics observed in Fig. 5 motivate the use of prediction methods that exploit early popularity measurements. Yet, this also raises a question as to whether the joint use of other features along with such measurements can improve prediction accuracy over exploiting only the latter (as in [35,38]). Similarly, the slow popularity evolution also raises a question as to how robust our solutions are to long-term predictions. We address these questions next.

### 6.2. *Prediction results varying the monitoring period* $\varepsilon$

We now investigate the accuracy of our prediction models for various values of the monitoring period ($\varepsilon$). We consider values of $\varepsilon$ equal to 1, 12, 24, 72 and 168 h (one week), fixing the target time $\delta$ equal to 1 month. Recall that, in these scenarios, we use as predictors all the features exploited in Section 5 as well as the number of likes already received by the tip $p_i$ and the position of $p_i$ in the ranking of tips posted at the venue sorted by date in descending order. For a given $\varepsilon$, the values of the predictors are computed taking all past history up to $\varepsilon$ hours after the tip's posting time.

Fig. 6 presents macro-averaged precision and recall results for each prediction method: SVM, SVR (with both linear and RBF kernels) and OLS. These results are produced using the complete set of features. The figure also shows results for the median baseline using all user features as well as only user/cat features (referred to as median-cat). Note that the results for $\varepsilon$=0 are the same results presented in Section 5.

We note that extending the monitoring time to only 1 h after the tip is posted only slightly improves the macro-averaged recall (up to 1.6% for SVM method), which is expected given the slow evolution of tip popularity observed. Yet, by monitoring the tip for one week ($\varepsilon$=168 h) we can improve the macro-averaged recall in up to 13% (SVM method) and the macro-average precision

**Table 3**
Macro-averaged results of models that use early popularity measurements (only tips with at least 1 like, $\varepsilon = 168$ h, $\delta = 1$ month).

| Metrics | Models | | | |
|---|---|---|---|---|
| | OLS | ML | MRBF | S–H |
| Macro-averaged recall | **0.8263** ± **0.0129** | 0.7257 ± 0.0064 | 0.8003 ± 0.0085 | **0.8395** ± **0.0100** |
| Recall low popularity | 0.8305 ± 0.0186 | **0.9960** ± **0.0005** | 0.9619 ± 0.0022 | 0.9240 ± 0.0010 |
| Recall high popularity | **0.8220** ± **0.018**9 | 0.4553 ± 0.0131 | 0.6386 ± 0.0176 | 0.7549 ± 0.0205 |
| Macro-averaged precision | 0.5650 ± 0.0061 | **0.8799** ± **0.0145** | 0.6674 ± 0.0129 | 0.6158 ± 0.0112 |
| Precision low popularity | **0.9930** ± **0.0012** | 0.9827 ± 0.0021 | 0.9879 ± 0.0017 | **0.9913** ± **0.0015** |
| Precision high popularity | 0.1369 ± 0.0126 | **0.7770** ± **0.0308** | 0.3469 ± 0.0271 | 0.2403 ± 0.0237 |

in up to 7% (SVR linear method) over using features computed at posting time. Moreover, such improvements are observed for all methods, although the median baselines are still much worse than our solutions. In particular, the OLS model remains as the most cost-effective prediction method as it produces results that are statistically as good as those obtained with the other (more costly) methods, for all considered values of $\varepsilon$.

Moreover, out of all considered features, the total number of likes received during the monitoring period $\varepsilon$ is the most discriminative feature according to the Information Gain criterion. This indicates the importance of taking the early popularity evolution as evidence for prediction. In other to assess to which extent the other features contribute to prediction accuracy, we also compare our OLS strategy against three other state-of-the-art baseline models that exploit only early popularity measurements.

The first one, proposed by Szabo and Huberman [38] and referred to as S–H model, is a univariate regression model that uses only the log-transformed total number of likes received during the monitoring period to predict the future (log-transformed) popularity. The other two baselines, proposed by Pinto et al. [35], are variations of the same method. One is a multivariate linear regression model that uses early popularity measures sampled at regular intervals (e.g., per day) during the monitoring period as predictors. The other builds on this model by also using Radial Basis Functions (RBFs) to capture the similarity (in the early popularity measurements) between the target tip and selected examples from the training set. These variations are referred to as ML and MRBF, respectively.[5] We use the S–H, ML and MBRF models as originally proposed, that is, to predict the total number of likes of a tip. We then use the predicted number to infer the corresponding popularity level.

We evaluate all models in the same scenario adopted by Szabo and Huberman [38] and Pinto et al. [35], that is, $\varepsilon$ equal to 168 h and $\delta$ equal to 1 month. As in Section 5, model parameters are defined using cross-validation in the training set. The number of RBF functions in the MRBF model was set to 100, as in [35].

We start by noting that although the S–H, ML and MRBF models can be directly applied to predict tip popularity, their use is constrained to tips with at least one like at the target time, since the models are solved by minimizing the mean relative squared error (MRSE) over the training set, which is undefined for tips with zero likes. Thus, in order to favor the baselines in our evaluation, we disregarded tips with zero likes, corresponding to 83% of all tips in our dataset.[6] Macro-averaged recall and precision results obtained with all models over the 17% remaining tips are shown in Table 3.[7]

Focusing first on recall, our primary metric of interest, we note that our OLS model produces the best recall results for the high popularity class, with gains of 80%, 29% and 9% over the ML, MRBF and S–H models, respectively. The baselines are more biased towards the less popular tips, favoring the recall of this class. Yet, in terms of macro-averaged recall, the OLS model still outperforms both ML and MRBF models, being statistically tied with the S–H model. The gains of OLS in recall, particularly for the high popularity class, come at the cost of a decrease in precision. The baselines, especially ML, are able to better filter false positives out, leading to higher precision.

In sum, we find that the baselines, particularly the ML model, perform quite well in the considered scenario, confirming previous results, except in terms of the recall of the high popularity class. As previously mentioned, this metric is of particular interest if one is aiming at retrieving most of the potentially more popular tips even if this comes at the expense of some noise. In such case, our OLS model is preferable.

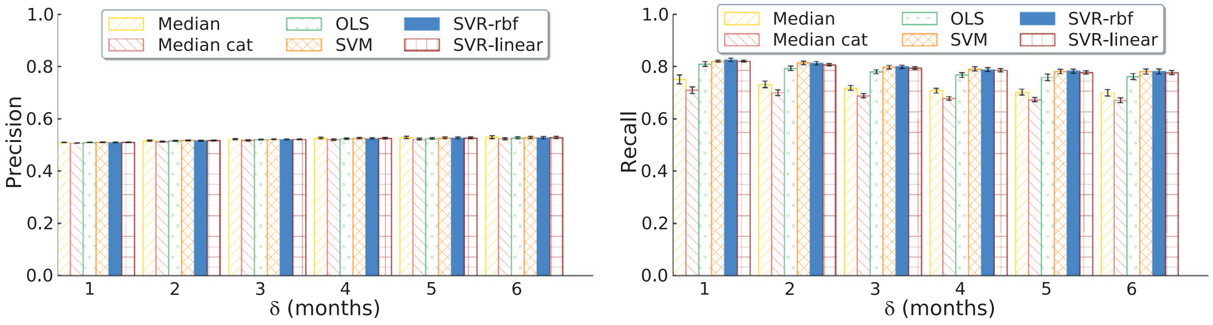
Moreover, we emphasize that our solution is more robust, since, unlike the baselines, it can be applied to any tip, at or after posting time. In particular, it can be applied to tips that have not received any like yet (i.e., unpopular tips or tips that have just been posted). The baselines, instead, are more suitable to types of content that exhibit a faster popularity evolution (e.g., news, videos).[8] Given the results discussed in Section 6.1, it is very likely that the baselines will not be applicable to the vast majority of the tips, even for other values of $\delta$. Our solution is thus more general and preferable.

[5] As extensions of the S–H model, both ML and MRBF also work on log-transformed popularity data.

[6] In this setup, the low popularity class is defined as tips with number of likes ranging from 1 to 4.

[7] Note that the OLS results shown in this table are different from those in Fig. 6, computed over all tips.

[8] Pinto et al. [35] reported that less than 1.5% of the videos in their datasets had not received any view in the first month in the system. This is in sharp contrast to 83% of the tips with no likes in the same period.



(a) Macro-Averaged Precision

(b) Macro-Averaged Recall

**Fig. 7.** Macro-averaged results for various target times $\delta$ ($\varepsilon = 0$).

### 6.3. *Prediction results varying target prediction window* $\delta$

We now analyze how prediction accuracy is affected as we vary the target prediction window $\delta$ from 1 to 6 months. In all scenarios, we set $\varepsilon$ to 0, thus focusing on predictions at posting time. We evaluate the OLS, SVM and SVR (with both kernels) models using all features. As baselines, we consider once again the median and median-cat models, since the S–H, ML and MRBF methods are not suitable to predictions at posting time.

Fig. 7 shows macro-averaged precision and recall results for all methods and values of $\delta$.[9] Note that the results for $\delta = 1$ are the same as those presented in Section 5. We find that, for any given value of $\delta$, OLS, SVR (with both kernels) and SVM produce statistically tied results in terms of macro-averaged precision (Fig. 7(a)). Moreover, the gains (if any) in macro-averaged recall (Fig. 7(b)) of the SVR and SVM methods over OLS are limited (up to 3.15% when $\delta = 5$ months). Thus, once again, OLS is a cost-effective solution, from a practical perspective, for various values of $\delta$.

Moreover, we find the same trend for all methods: as $\delta$ increases, precision increases slightly but at the cost of a (small) reduction on recall. For example, comparing the predictions done by OLS for $\delta$ equal to 1 and 6 months, we observe a small improvement in macro-averaged precision (3.37%) for the latter, but also a decrease in macro-averaged recall (5.9%). As shown in Fig. 5(b), for a large fraction of tips, most likes are gained after 3 months, which suggests that a tip may take a long time to converge to its final popularity level. Yet, we still observe a loss in macro-averaged recall of 3.7% when predictions are done for $\delta$=3. The losses in recall occur in both classes, reaching 6.35% and 5.47% for low and high popularity classes respectively, for $\delta$ equal to 6.

The gains in macro-averaged precision observed as $\delta$ increases come from a higher precision for high popularity class, which, in turn, is due to the reduction of class imbalance (which severely hurts the precision of the smaller class) as more tips migrate from the low to the high popularity class. This migration might also partially explain the losses in recall. More generally, as we predict further ahead, model inputs (feature values) become outdated and less efficient for prediction purposes. Given our results, we find that predictions for up to 2 months ahead are mostly unaffected by outdated features. For longer periods, the reduction in recall starts becoming significant.

## 7. Model specialization (Q3)

So far, we have built and evaluated prediction models that were trained and tested using *all* tips in the dataset. This approach produces a single general prediction model that aggregates and summarizes the relationships between the predictors (feature values) and the response (popularity level) across all tips. In this section, we analyze whether we can improve prediction accuracy by building models that are specialized to particular groups of tips such as tips posted at venues located in the same geographic region or venues of the same category. Model specialization might bring up patterns that are inherent to that particular group of tips, but are masked when all tips are treated jointly. For example, venues in different categories might exhibit different patterns: while “Travel & Transport” is the most popular venue category in terms of number check-ins, “Food” is the category that attracts the largest number of tips [25]. Model specialization might improve accuracy as fewer instances of noise are used to train the models. On the other hand, specialization might also suffer from the lack of enough training instances, which impacts prediction accuracy as it affects the capacity of the model to generalize, or from a more severe class imbalance which, due to the need of undersampling in the training set, ends up severely restricting the amount of training examples.

We here assess the benefits from building specialized models for specific cities (Section 7.1) and venue categories (Section 7.2). To that end, we compare specialized and general models, built using the same method (OLS, SVM or SVR) on the *same* test set, when performing predictions at posting time for one month in the future (i.e., $\varepsilon$=0 and $\delta$=1 month). We adopt the same general

[9] The tip distribution across classes may not be the same for experiments with different values of $\delta$ since tips from the low popularity class may move to the high popularity class as $\delta$ increases.

experimental setup described in Section 5.1, learning model parameters through cross-validation in the training set. There are two key differences though. First, the set of tips used as input to the experimental procedure is restricted to tips posted in venues of the target city or category in case of specialization. Second, when training either the general or the specialized models, we here consider multiple tips posted by the same user as candidates for prediction, since, using only the most recent tip of each user, as discussed in Section 5.1, severely constrains the amount of data available for training.

### *7.1. City-based model specialization*

We start by assessing the benefits of building specialized models for specific cities. To that end, we build models using tips posted at venues located in four selected cities, namely New York (NY), Los Angeles (LA), San Francisco (SF), and Chicago (CHI).[10] For each city, we compare the specialized model against a general model using the same test set composed of only tips posted at venues of the target city. Moreover, for a fair comparison, the general model is built using a training set of size equal to the one used to learn the specialized model, although the former consists of tips posted in venues of all cities in the dataset, randomly sampled from the original (global) training set. Specifically, the training sets used to learn the models for the NY, LA, SF and CHI scenarios contain 1141, 293, 308, and 183 tips, respectively. Similarly, the learned models were applied on test sets including 5085, 1551, 1695 and 1443 tips, respectively.

Table 4 shows macro-averaged recall and precision results, along with corresponding 95% confidence intervals, for each model and method. For each scenario (city), the best methods (including statistical ties) are shown in bold. A ↑ (or ↓) sign is used to indicate a statistical improvement (or loss) of the specialized model over the corresponding general model. The lack of a sign indicates a statistical tie.

We observe that the specialized models outperform (or at least are statistically tied with) the corresponding general models in the vast majority of the cases. The improvements occur particularly in terms of macro-averaged precision, varying from 1.60% to 3.76%. Such improvements in precision occur in the high popularity class. Indeed, the gains in average precision for that class achieve 84.87% for the SF scenario with SVR (with RBF kernel) model. In terms of macro-averaged recall, both specialized and general models are statistically tied. Moreover, unlike observed in the previous sections, we do find cases where SVM and SVR (with RBF kernel) significantly outperform the simpler OLS in terms of macro-averaged precision. Note, for example, the difference between these methods for SF, CHI and LA. Such gains are not directly related to the specialization, but rather to the greater robustness of SVM and SVR to smaller training sets [35]. Indeed such gains are observed for both general and specialized models.

In sum, we find that city-based model specialization does bring some improvements, particularly in terms of precision, as specialized models are able to more accurately capture patterns that are specific to the target city, reducing the amount of false negatives. One point to note, though, is that the amount of information available to learn a specialized model is inevitably smaller, if compared to a general model, and may require the use of techniques that are more robust to the lack of training instances.

As a final observation in this analysis, we note that by evaluating specialized models for each city, we are also analyzing the benefits from adding spatial (i.e., geographic) factors to our prediction models. We consider spatial factors at the city level because, on Foursquare, the geographic information associated with each user, one of the central entities of the popularity prediction problem, is available only at the city level.

We also note that spatial information at a finer granularity (i.e., latitude and longitude coordinates) is available only for venues, and our models do capture patterns associated with particular venues by taking venue-specific features into account. The city-based model specialization captures new factors that may exist due to spatial locality of tip popularity patterns at the city level. We leave to future work a more thorough investigation of other spatial factors as well as other strategies to introduce them to the popularity prediction models.

### *7.2. Category-based model specialization*

Finally, we analyze the benefits of building specialized models for each venue category. Recall that Foursquare defines nine top level venue categories, namely, “Arts & Entertainment”, “Colleges & Universities”, “Food”, “Great Outdoors”, “Nightlife Spots”, “Travel and Transport”, “Shops”, “Professional & Other Places” and “Residences”. We build specialized models for four selected categories, namely “Arts & Entertaining” (Arts), “Food”, “Shops & Services” (Shops), and “Nightlife Spots” (Night).[11] As before, we compare each specialized model with the corresponding general model (considering all categories) on the same test set of tips posted at venues of the target category. Moreover, both models are learned with training sets of the same size. Specifically, the training sets used to learn the models for the Arts, Food, Shops and Night scenarios consist of 1390, 2093, 1326, and 1086 tips, respectively, whereas corresponding test sets include 5790, 49,341, 17,965, and 9685 tips.

Table 5 shows macro-averaged recall and precision results, along with corresponding 95% confidence intervals, for each model and method. Like in Table 4, results for the best methods (including statistical ties) for each category are shown in bold, and ↑ and ↓ signs are used to indicate gains and losses due to specialization.

Our first observation is that category-based model specialization does not bring as large and clear improvements over the general model as the city-based specialization. On one hand, we do observe some statistical improvements in macro-averaged

[10] These cities were selected as they have the largest number of tips in our dataset.

[11] These categories were selected as they have the largest numbers of tips in our dataset.

**Table 4**
City-based model specialization: macro-averaged results.

| Scenarios | Macro-averaged recall | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | General model | | | | Specialized model | | | |
| | OLS | SVM | SVR-RBF | SVR-linear | OLS | SVM | SVR-RBF | SVR-linear |
| NY | **0.8123 ± 0.0290** | **0.8175 ± 0.0278** | **0.8286 ± 0.0253** | **0.8267 ± 0.0277** | **0.8147 ± 0.0169** | **0.8099 ± 0.0176** | **0.8242 ± 0.0189** | **0.8069 ± 0.0287** |
| SF | 0.7817 ± 0.0271 | **0.8395 ± 0.0313** | **0.8520 ± 0.0197** | **0.8519 ± 0.0167** | 0.8041 ± 0.0430 | **0.8315 ± 0.0486** | **0.8185 ± 0.0556** | **0.8177 ± 0.0465** |
| CHI | 0.7379 ± 0.0295 | **0.8038 ± 0.0370** | **0.8161 ± 0.0328** | **0.8188 ± 0.0337** | 0.7206 ± 0.0476 | **0.7701 ± 0.050**1 | **0.7767 ± 0.0504** | **0.7734 ± 0.0506** |
| LA | 0.7860 ± 0.0309 | **0.8421 ± 0.0314** | **0.8545 ± 0.0313** | **0.8638 ± 0.0326** | 0.7983 ± 0.0504 | **0.8489 ± 0.0383** | **0.8594 ± 0.0326** | **0.8540 ± 0.0330** |
| NY | 0.5336 ± 0.0067 | 0.5345 ± 0.0072 | 0.5332 ± 0.0075 | 0.5372 ± 0.0083 | **0.5537 ± 0.0090** ↑ | **0.5499 ± 0.0081** ↑ | **0.5485 ± 0.0091** ↑ | 0.5429 ± 0.0093 |
| SF | 0.5121 ± 0.0022 | 0.5204 ± 0.0065 | 0.5201 ± 0.0053 | 0.5244 ± 0.0074 | 0.5221 ± 0.0040 ↑ | **0.5338 ± 0.0070** ↑ | **0.5369 ± 0.0087** ↑ | **0.5369 ± 0.0094** ↑ |
| CHI | 0.5098 ± 0.0015 | 0.5158 ± 0.0034 | 0.5163 ± 0.0031 | 0.5189 ± 0.0036 | 0.5118 ± 0.0023 | **0.5259 ± 0.0051** ↑ | **0.5269 ± 0.0046** ↑ | **0.5272 ± 0.0050** ↑ |
| LA | 0.5169 ± 0.0035 | 0.5242 ± 0.0049 | 0.5260 ± 0.0051 | 0.5314 ± 0.0064 | 0.5265 ± 0.0072 ↑ | **0.5407 ± 0.0078** ↑ | **0.5426 ± 0.0078** ↑ | **0.5423 ± 0.0083** ↑ |

**Table 5**
Categorical model specialization: macro-averaged results.

| Scenarios | Macro-averaged recall | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | General model | | | | Specialized model | | | |
| | OLS | SVM | SVR-RBF | SVR-linear | OLS | SVM | SVR-RBF | SVR-linear |
| Arts | **0.8010 ± 0.0132** | **0.7865 ± 0.0139** | **0.7959 ± 0.0153** | **0.7848 ± 0.0371** | **0.7832 ± 0.0108 ↓** | **0.7937 ± 0.0124** | **0.7896 ± 0.0112** | **0.7795 ± 0.0351** |
| Food | 0.7884 ± 0.0193 | 0.7972 ± 0.0188 | 0.7853 ± 0.0196 | 0.7682 ± 0.0153 | **0.8140 ± 0.0164 ↑** | **0.8250 ± 0.0153 ↑** | **0.8262 ± 0.0137 ↑** | 0.7627 ± 0.0096 |
| Night | 0.7921 ± 0.0102 | **0.8048 ± 0.0079** | **0.8036 ± 0.0081** | 0.7707 ± 0.0167 | **0.8065 ± 0.0113 ↑** | **0.8134 ± 0.0102** | 0.8008 ± 0.0086 | 0.7627 ± 0.0075 |
| Shops | **0.8119 ± 0.0347** | **0.8130 ± 0.0310** | **0.8124 ± 0.0337** | 0.7042 ± 0.0359 | **0.8187 ± 0.0331** | **0.8156 ± 0.0353** | **0.8092 ± 0.0355** | 0.7609 ± 0.0366 ↑ |
| Arts | 0.5279 ± 0.0059 | 0.5237 ± 0.0058 | 0.5255 ± 0.0065 | 0.5627 ± 0.0026 | 0.5470 ± 0.0076 ↑ | 0.5497 ± 0.0089 ↑ | 0.5494 ± 0.0095 ↑ | **0.5658 ± 0.0025 ↑** |
| Food | 0.5258 ± 0.0036 | 0.5288 ± 0.0048 | 0.5271 ± 0.0053 | 0.5234 ± 0.0034 | 0.5194 ± 0.0026 ↓ | 0.5186 ± 0.0027 ↓ | 0.5212 ± 0.0040 ↓ | **0.5360 ± 0.0047 ↑** |
| Night | 0.5256 ± 0.0049 | 0.5275 ± 0.0050 | 0.5314 ± 0.0064 | **0.5450 ± 0.0067** | 0.5292 ± 0.0032 | 0.5328 ± 0.0047 | **0.5383 ± 0.0066** | **0.5448 ± 0.0071** |
| Shops | 0.5232 ± 0.0042 | 0.5244 ± 0.0047 | 0.5246 ± 0.0054 | **0.5285 ± 0.0094** | 0.5271 ± 0.0041 | 0.5261 ± 0.0044 | 0.5261 ± 0.0044 | **0.5363 ± 0.0061** |

recall and precision of up to 8.06% and 4.98%, respectively. Yet, statistical losses in the same metrics of up to 2.22% and 1.92% are also observed. Overall, we find that the results of the specialized models are only marginally different (if not tied with) those produced by the corresponding general models.

Such small differences are mainly due to the fact that the venue category is already somewhat explored by our (general) model as a feature. Indeed, as discussed in Section 5.2.2, venue category is one of the top 10 most important features for popularity prediction, implying that different patterns may exist across different categories. Yet, since this feature is already part of the general model, specialization for each category does not bring as much new information to the model as the city-based specialization does (which, as discussed, introduces factors related to city-level spatial locality to the model). This is why we observed a tie between general and specialized models in various scenarios. The statistically significant differences (improvements and losses) observed in a few cases (e.g., OLS on Food and Arts categories) are caused by differences in the training set distributions used to build both general and specialized models. These differences in turn are an indirect result of the specialization, as we further explain next.

Recall that in our experiments, for a given category, the training sets used to learn both specialized and general models have the same number of tips. However, the training set of the general model contains tips from all categories. Thus, if we consider only tips of the target category, the number of tips in each class (and the class imbalance) may be different in both training sets.

Take, for example, the case of the Food category. Table 5 shows that, for OLS, SVM and SVR (with RBF kernel), the specialization does bring some improvements in macro-averaged recall but losses in macro-averaged precision. We manually investigated the tips in the training sets (some folds) used to build both models. Focusing only on tips in Food venues, the training set of the specialized model includes a proportionally much larger number of tips of high popularity than the training set of the general model. This favors the classification of tips in the high popularity class by the specialized model, which leads to improvements in recall for that particular class. However, as a side effect, the number of tips in the low popularity class that are incorrectly classified also increases, which leads to losses in precision (once again for the high popularity class, as it is smaller and more sensitive to changes).

For the Arts category, we observe an opposite trend: specialization leads to losses in macro-averaged recall but improvements in macro-averaged precision. Once again, this can be explained by the different numbers of tips in each class in both training sets, considering only venues in the Arts category. Compared to the training set of the general model, the training set of the specialized model includes a proportionally much larger number of tips of the low popularity class. This favors the classification of tips in that class, which hurts recall of the high popularity class but also improves its precision.

We note that different techniques (OLS, SVM, SVR) may be more or less robust to such differences in training set. We also emphasize that it was not expected beforehand that the category-based model specialization would not bring consistent improvements over the general model, despite the latter including venue category as a feature. The two prediction models exploit venue category very differently. In the general model, this information is another dimension of the feature space considered. Given the high dimensionality of this space (125 features), using only tips of the target category to learn the model could help to reduce noise and significantly improve prediction accuracy. Yet, our results revealed that the specialization is not consistently beneficial and may also hurt prediction accuracy.

Thus, given such inconsistent performance, and considering that the differences, when significant, are small, we argue that category-based model specialization is not worthwhile because the main additional information, venue category, is already considered by the general model (though in a different way).

As a final observation, we also note that the gains of the more sophisticated SVM and SVR over OLS are not as large as observed for the city-based specialization. They are constrained to at most 3.77% and 3.44% for macro-averaged recall and precision, respectively, which limit their benefits over the simpler OLS, from a practical perspective. The limited benefits of SVM and SVR are probably due to the amounts of training examples available in the category-based scenarios, which unlike in some of the city-based scenarios, are large enough for OLS to produce reasonably accurate results.

## 8. Conclusions and future work

We have tackled the problem of predicting the future popularity level of micro-reviews (or tips) on Foursquare. Our investigation was driven by three research questions: we analyzed the most important factors for predicting the popularity of Foursquare tips as soon as it is posted (Q1); we investigated the extent to which the monitoring time and the target prediction window affect prediction accuracy (Q2), and we assessed the benefits from city and venue category based model specialization (Q3). To the best of our knowledge, this is the first time that the popularity or helpfulness prediction of micro-reviews is addressed by exploiting content, temporal, spatial, topical and, social aspects of the problem in a conjoint way.

Specifically, we have proposed a rich set of features capturing factors that may influence tip popularity. We have also evaluated state-of-the-art classification and regression-based strategies to predict the popularity level of a tip at a future time, using various subsets of the proposed features as predictor variables and considering various prediction scenarios. Our experimental results showed that the simpler OLS algorithm, using features of both the user who posted the tip and the venue where it was posted as predictors, produces results that are statistically as good as (if not better than) those obtained with the more sophisticated, but also most costly, SVM and SVR methods in most cases. Nevertheless, SVM and SVR are more robust solutions if the amount of data available for training the prediction model is too small. Moreover, our results also showed that the use of only the top-10 most discriminative features, which include features that capture the prior popularity of the user who posted the tip and the

**Table 6**
Complete set of features used by the our popularity prediction models.

| Type | Feature ID | Description |
|---|---|---|
| User | 1 | Total number of tips posted by the user |
| | 2 | Number of venues where the author posted tips |
| | [3–6] | Total number of likes received by previous tips of the author[a] |
| | 7 | Total number of likes given by the tip's author |
| | 8 | Number of friends or followers of the author |
| | 9 | Ratio of all likes received by the author coming from his friends and followers |
| | [10–13] | Total number of tips posted by the author's social network[a] |
| | [14–17] | Number of likes given by author's social network (in any tip)[a] |
| | [18–21] | Fraction of all likes received by the tip's author that are associated with tips posted at the same venue of the current tip but after it was posted[a] |
| | 22 | User category defined by Foursquare |
| | 23 | Total number of mayorships won by the author |
| | 24 | If the author was mayor of the venue where tip was posted |
| Venue | 25 | Total number of tips posted at the venue |
| | [26–29] | Total number of likes received by tips posted at the venue[a] |
| | 30 | Total number of check ins at the venue |
| | 31 | Total number of unique visitors |
| | 32 | If the tipped venue was verified by Foursquare |
| | 33 | Venue category defined by Foursquare |
| | 34 | Position of the tip in the tips of the venue sorted by # of likes in ascending order |
| | 35 | Position of the tip in the ranking of the venue sorted by # of likes in descending order |
| | 36 | Position of the tip in the ranking of the venue sorted by date in ascending order |
| | 37 | Position of the tip in the ranking of the venue sorted by date in descending order [b] |
| Content | 38 | Number of likes received by the tip during monitoring time (i.e., until time $tp_i + \epsilon$) [b] |
| | 39 | Length of the text of the tip, in characters |
| | 40 | Length of the text of the tip, in number of words |
| | 41 | Number of URLs or e-mails address contained on a tip |
| | 42 | Fraction of nouns in the tip |
| | 43 | Fraction of adjectives in the tip |
| | 44 | Fraction of adverbs in the tip |
| | 45 | Fraction of comparatives in the tip |
| | 46 | Fraction of verbs in the tip |
| | 47 | Fraction of non-English words in the tip |
| | 48 | Fraction of numbers in the tip |
| | 49 | Fraction of superlatives in the tip |
| | 50 | Fraction of symbols in the tip |
| | 51 | Fraction of punctuation in the tip |
| | 52 | Average positive score over all words in the tip |
| | 53 | Average neutral score over all words in the tip |
| | 54 | Average negative score over all words in the tip |
| | 55 | Fraction of new tips terms with respect of the other tips in the same venue |
| | 56 | Fraction of unique words |
| | 57 | Fraction of capitalized words in the tip |
| | 58 | Number of bad words in the tip[c] |
| | 59 | Number of words present in a list of common misspellings[d] |
| | 60 | Number of words in the tip that are not in the WordNet |
| | 61 | Total number of syllables in the tip text |
| | 62 | Average number of syllables per tip word |
| | 63 | Average number of words per tip sentence |
| | 64 | Average number of characters per word |
| | 65 | Number of words in the largest sentence |
| | 66 | Number of words with 3 or more syllables |
| | 67 | Ratio of number of spaces in the tip |
| | 68 | Number of sentences in the tip text |
| | 69 | Entropy of the tip word sizes |
| | 70 | Number of sentences in the tip text beginning with a conjunction |
| | 71 | Number of sentences in the tip text beginning with an article |
| | 72 | Number of sentences in the tip text beginning with an interrogative pronoun |
| | 73 | Number of sentences in the tip text beginning with a preposition |
| | 74 | Number of sentences in the tip text beginning with a pronoun |
| | 75 | Number of sentences in the tip text beginning with a subordinating conjunctions |
| | 76 | Number of uses of verb "to be" |
| | 77 | Number of pronouns in the tip text |
| | 78 | Number of passive voice sentences |
| | 79 | Number of prepositions in the tip text |
| | 80 | Number of question marks in the tip text |
| | 81 | Number of conjunctions in the tip text |
| | 82 | Number of entities mentioned in the tip |
| | 83 | Number of distinct types of named entities mentioned in the tip |
| | [84–89] | Readability scores (Automated Readability Index, Flesch Reading ease, SMOG-Grading, Flesh-Kincaid, Coleman-Liau, Gunning Fog) |
| | [90–125] | Fraction of words in the tip that belong to each of the LIWC class |

[a] Median, average and standard deviation are also included. Feature IDs are assigned in sequence: total, median, average and standard deviation.

[b] This feature is used only when $\epsilon > 0$.

[c] Based on the list published in https://gist.github.com/jamiew/1112488.

[d] Based on the list published in http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings.

venue where it was posted as well as characteristics of the user's social network, produces results that are as good as when all 125 features are used as predictors.

We found significant improvements for all prediction methods when extending the monitoring time. Moreover, although state-of-the-art prediction methods that use early popularity measurements as the only predictors do perform reasonably well in such scenarios, our models are more robust, as they can be applied to any tip, at or after posting time (unlike the other methods), besides producing much higher recall for the high popularity class. We also found small improvements in precision, particularly for the high popularity class, as we increase the target prediction window, mostly due to a reduction in class imbalance. However, such improvements come at the cost of a reduction in recall, particularly if predictions are performed for more than 2 months ahead in the future. The reason for such reduction is that, as we predict further into the future, model inputs become outdated and less efficient for prediction purposes. Finally, we found that model specialization does bring some improvements if performed at the city-level, as it captures specific patterns that may exist in the geographic region where the city is located. Category-based specialization, on the other hand, does not bring clear and consistent gains, because the venue category is already somewhat exploited by the general model as a predictor variable.

Predicting the popularity of micro-reviews (tips in particular) is a challenging problem which, in comparison with previous related efforts, has unique aspects and inherently different characteristics, and may depend on a non-trivial combination of various user, venue and content features. We expect that the knowledge derived from the present effort may bring valuable insights into the design of more cost-effective automatic tip filtering and recommendation strategies. Other popularity prediction tasks, such as predicting the exact number of likes a tip will receive at a future time, or predicting the popularity ranking of a group of tips, are also worth pursuing.

In addition, even though this study is focused on Foursquare as a case study, we do expect that many of our general findings, which are related to the target problem, still hold in other micro-review systems. For instance, even though features might need to be adapted to the particular application, we do expect that user and venue features are very important to popularity prediction while content features play a less preeminent role. We also expect a good performance of the simpler OLS as well as improvements from predicting after a monitoring time. Obviously, these are only conjectures but the current results motivate future studies to validate our findings in other similar systems.

Finally, another direction worth pursuing as future work is the investigation of other strategies to capture spatial factors in our prediction models. A characterization of the spatial locality of tip popularity could produce valuable insights into how to capture such factors.

## Acknowledgment

This research is partially funded by the Brazilian National Institute of Science and Technology for Web Research (MCT/CNPq/INCT Web Grant number 573871/2008-6), and by the authors' individual grants from CNPq, CAPES and FAPEMIG.

## Appendix

Table 6 presents the complete list of features used by our popularity prediction models. Note that, in some cases, different variations of the same feature are included, such as total, median, average and standard deviation of the number of likes received by previous tips of the author (features 3–6).

We also emphasize that features 2, 7, 9, 10–24, 30–31 have not been explored before, being thus new contributions of this work. Other features, notably features 32, 33, 38, 41, 55, 60, 65, 67, 69–71, 73–79, 81, 90–125 have been adapted from similar concepts in other domains (e.g., question-answering sites, Wikipedia, and YouTube) [1,11,12,30,36].

## References

[1] A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, Discovering Value from Community Activity on Focused Question Answering Sites: a Case Study of Stack Overflow, in: International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2012, pp. 850–858.

[2] R. Bandari, S. Asur, B. Huberman, The Pulse of News in Social Media: Forecasting Popularity, in: International Conference on Weblogs and Social Media (ICWSM), 2012, pp. 26–33.

[3] Y. Borghol, S. Ardon, N. Carlsson, D. Eager, A. Mahanti, The Untold Story of the Clones: Content-agnostic Factors That Impact YouTube Video Popularity, in: International Conference on Knowledge Discovery and Data Mining (KDD), 2012, pp. 1186–1194.

[4] C. Castillo, M. El-Haddad, J. Pfeffer, M. Stempeck, Characterizing the Life Cycle of Online News Stories Using Social Media Reactions, in: Conference on Computer Supported Cooperative Work & Social Computing (CSCW), 2014, pp. 211–223.

[5] M. Cha, A. Mislove, K. Gummadi, A Measurement-driven Analysis of Information Propagation in the Flickr Social Network, in: International Conference on World Wide Web (WWW), 2009, pp. 721–730.

[6] C. Chang, C. Lin, LIBSVM: a Library for Support Vector Machines, (2001).

[7] B.-C. Chen, J. Guo, B. Tseng, J. Yang, User Reputation in a Comment Rating Environment, in: International Conference on Knowledge Discovery and Data Mining (KDD), 2011, pp. 159–167.

[8] P. Chen, S. Wu, J. Yoon, The Impact of Online Recommendations and Consumer Feedback on Sales, in: International Conference on Information Systems (ICIS), 2004, pp. 711–724.

[9] Chiao-Fang, E. Khabiri, J. Caverlee, Ranking Comments on the Social Web, in: International Conference on Computational Science and Engineering (CSE), 2009, pp. 90–97.

[10] R. Crane, D. Sornette, Robust Dynamic Classes Revealed by Measuring the Response Function of a Social System, in: PNAS, 2008, pp. 15649–15653.

[11] D. Dalip, M. Gonçalves, M. Cristo, P. Calado, Automatic Assessment of Document Quality in Web Collaborative Digital Libraries, JDIQ 2 (3) (2011) 14:1–14:30.

[12] D. Dalip, M. Gonçalves, M. Cristo, P. Calado, Exploiting User Feedback to Learn to Rank Answers in Q&A Forums: A Case Study with Stack Overflow, in: International Conference on Research and Development in Information Retrieval (SIGIR), 2013, pp. 543–552.

[13] A. Esuli, F. Sebastiani, SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining, in: Conference on Language Resources and Evaluation (LREC), 2006, pp. 417–422.

[14] A. Ghose, P. Ipeirotis, Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics, IEEE TKDE 23 (10) (2011) 1498–1512.

[15] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer, 2008.

[16] H. He, E. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009).

[17] L. Hong, O. Dan, B. Davison, Predicting popular messages in Twitter, in: Proceedings of the International Conference on World Wide Web (WWW), 2011, pp. 57–58.

[18] Y. Hong, J. Lu, J. Yao, Q. Zhu, G. Zhou, What reviews are satisfactory: novel features for automatic helpfulness voting, in: Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR), 2012, pp. 495–504.

[19] T. Howley, M. Madden, M. O'Connell, A. Ryder, The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data, Knowledge-Based Syst. 19 (5) (2006) 363–370.

[20] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley, 1991.

[21] S. Kim, P. Pantel, T. Chklovski, M. Pennacchiotti, Automatically assessing review helpfulness, in: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2006, pp. 423–430.

[22] T. Lappas, Fake reviews: the malicious perspective, Lecture Notes in Computer Science (NLDB), Springer Berlin Heidelberg, 2012, pp. 23–34.

[23] S. Lee, J. Choeh, Predicting the helpfulness of online reviews using multilayer perceptron neural networks, Expert Syst. Appl. 41 (6) (2014) 3041–3046.

[24] B. Li, T. Jin, M. Lyu, I. King, B. Mak, Analyzing and predicting question quality in community question answering services, in: Proceedings of International Conference on World Wide Web (WWW), 2012, pp. 775–782.

[25] Y. Li, M. Steiner, L. Wang, Z.-L. Zhang, J. Bao, Exploring venue popularity in Foursquare, in: Proceedings of IEEE INFOCOM Workshops, 2013, pp. 205–210.

[26] Y. Lin, T. Zhu, X. Wang, J. Zhang, A. Zhou, Towards online review spam detection, in: Proceedings of International World Wide Web Conference (WWW), 2014, pp. 341–342.

[27] J. Liu, Y. Cao, C. Lin, C. Lin, Y. Huang, M. Zhou, Low-quality product review detection in opinion summarization, in: Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 334–342.

[28] Y. Liu, X. Huang, A. An, X. Yu, Modeling and predicting the helpfulness of online reviews, in: Proceedings of IEEE International Conference on Data Mining (ICDM), 2008, pp. 443–452.

[29] Y. Lu, P. Tsaparas, A. Ntoulas, L. Polanyi, Exploiting social context for review quality prediction, in: Proceedings of International Conference on World Wide Web (WWW), 2010, pp. 691–700.

[30] E. Momeni, C. Cardie., M. Ott, Properties, prediction, and prevalence of useful user-generated comments for descriptive annotation of social media objects, in: Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM), 2013, pp. 390–399.

[31] R. O'Brien, A caution regarding rules of thumb for variance inflation factors, Quality & Quantity: International Journal of Methodology 41 (5) (2007) 673–690.

[32] M. O'Mahony, B. Smyth, Learning to Recommend Helpful Hotel Reviews, in: Proceedings of Conference on Recommender Systems (RecSys), 2009, pp. 305–308.

[33] M. O'Mahony, B. Smyth, Using readability tests to predict helpful product reviews, in: Proceedings of Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO), 2010, pp. 164–167.

[34] B. Pang, L. Lee, S. Vaithyanathan, Thumbs Up? Sentiment classification using machine learning techniques, in: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.

[35] H. Pinto, J. Almeida, M. Gonçalves, Using early view patterns to predict the popularity of YouTube videos, in: Proceedings of International Conference on Web Search and Data Mining (WSDM), 2013, pp. 365–374.

[36] S. Siersdorfer, S. Chelaru, W. Nejdl, J. San Pedro, How useful are your comments?: Analyzing and predicting YouTube comments and comment ratings, in: Proceedings of International Conference on World Wide Web (WWW), 2010, pp. 891–900.

[37] J. Stevens, Applied Multivariate Statistics for the Social Sciences, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2002.

[38] G. Szabo, B. Huberman, Predicting the popularity of online content, Commun. ACM 53 (8) (2010) 80–88.

[39] A. Tatar, P. Antoniadis, M. Amorim, S. Fdida, From popularity prediction to ranking online news, Soc. Netw. Anal. Min. 4 (1) (2014).

[40] R. Tausczik, J. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods, J. Lang. Soc. Psychol. 29 (1) (2010) 24–54.

[41] T. Ngo-Ye, A. Sinha, The influence of reviewer engagement characteristics on online review helpfulness: a text regression model, Decis. Support Syst. 61 (2014) 47–58.

[42] M. Vasconcelos, J. Almeida, M. Goncalves, Popularity dynamics of Foursquare micro-reviews, in: Proceedings of Conference on Online Social Networks (COSN), 2014, pp. 119–130.

[43] M. Vasconcelos, J. Almeida, M. Goncalves, What makes your opinion popular? Predicting the popularity of micro-reviews in Foursquare, in: Proceedings of Annual ACM Symposium on Applied Computing (SAC), 2014, pp. 598–603.

[44] M. Vasconcelos, S. Ricci, J. Almeida, F. Benevenuto, V. Almeida, Tips, dones and todos: uncovering user profiles in Foursquare, in: Proceedings of International Conference on Web Search and Data Mining (WSDM), 2012, pp. 653–662.

[45] Z. Zhang, B. Varadarajan, Utility scoring of product reviews, in: Proceedings of International Conference on Information and Knowledge Management (CIKM, 2006, pp. 51–57.

[46] D. Zwillinger, S. Kokoska, CRC Standard Probability and Statistics Tables and Formulae, 1st edition, Chapman & Hall/CRC, 2000.

**Marisa A. Vasconcelos** received her Ph.D., M.Sc. and B.S. degrees in Computer Science from Universidade Federal de Minas Gerais (UFMG), Brazil in 2015, 2003 and 2000, respectively. She received also a M.A. degree also in Computer Science in 2008 from Boston University, US. She is a recipient of a Fulbright award for her M.A. studies and also a recipient of Google Women in Technology (Anita Borg) award. Her research interests are in the areas of user behavior characterization, social computing, computational social science, and prediction analysis.

**Jussara M. Almeida** holds Ph.D. and M.Sc. degrees in Computer Science by the University of Wisconsin-Madison, US (2003 and 1999, respectively), as well as a Master and Bachelor degrees also in Computer Science by the Universidade Federal de Minas Gerais (UFMG), Brazil (1997 and 1994, respectively). She is currently an Associate Professor in Computer Science at UFMG as well as Affiliated Member of the Brazilian Academy of Science. Her research interests include performance modeling and analysis of large-scale distributed systems, workload and user behavior modeling of large-scale distributed systems, social computing and recommendation systems.

**Marcos André Gonçalves** is an associate professor of computer science at the Universidade Federal de Minas Gerais, Brazil. He received the Ph.D. degree in Computer Science from Virginia Tech (2004). His research interests include information retrieval, digital libraries, text classification, and text mining in general.